Software quality

General concepts

What is software quality?

There is NOT a unique answer

What are the characteristics
for high quality software?

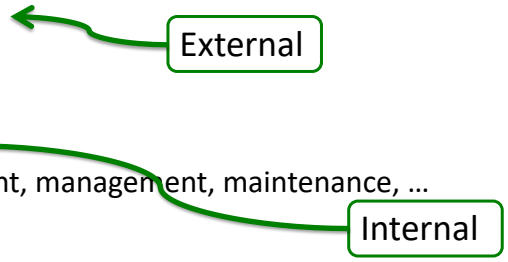
- Different perspectives
- Different expectations

Different views

- Transcendental: no definition
  - … quality is recognized if present!
- User: meeting users' needs
- Manufacturing: conformance to process standards
- Product: focus on internal characteristics
- Value-based: willingness to pay for the product

Another view classification

- Different people have different views
  - Role
  - Responsibilities
- People can be divided in two groups:
  - Consumers
    - Customers
    - Users
  - Producers
    - Development, management, maintenance, …



Which are your expectations as a consumer?

Consumer view

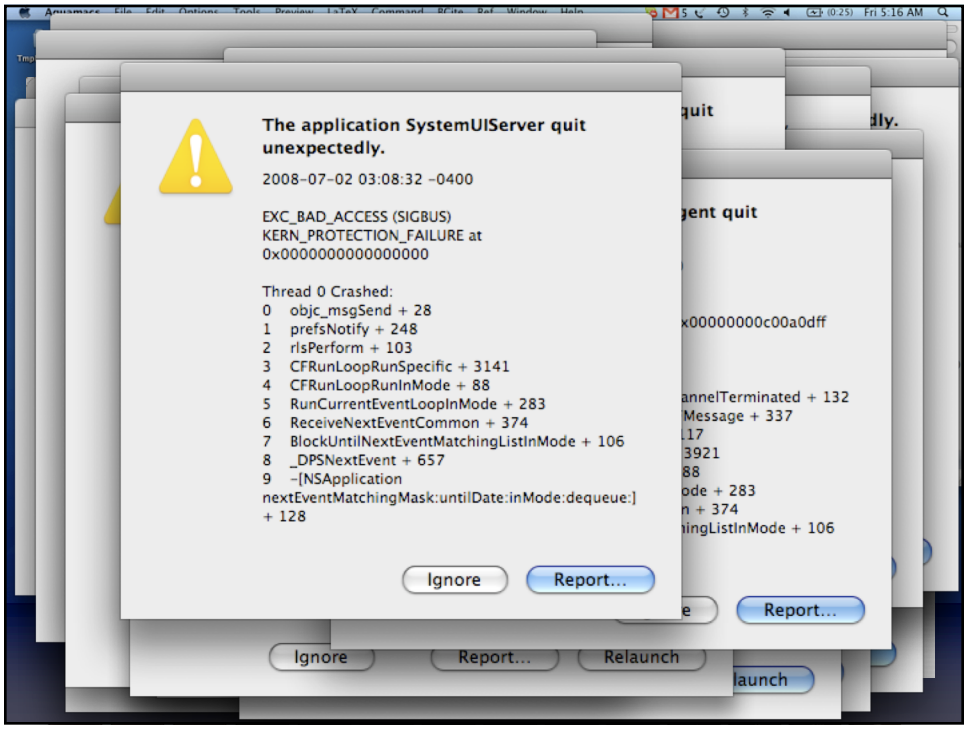
- Useful functions as specified
  - Fit user's needs
  - Functions pare performed reliably
- "Easy to use" and "easy to install"
  - number of steps
  - types of interfaces
  - time to learn/figure-out
  - …
- Interoperability and adaptability
- Cost reasonable w.r.t the offered functionalities/services

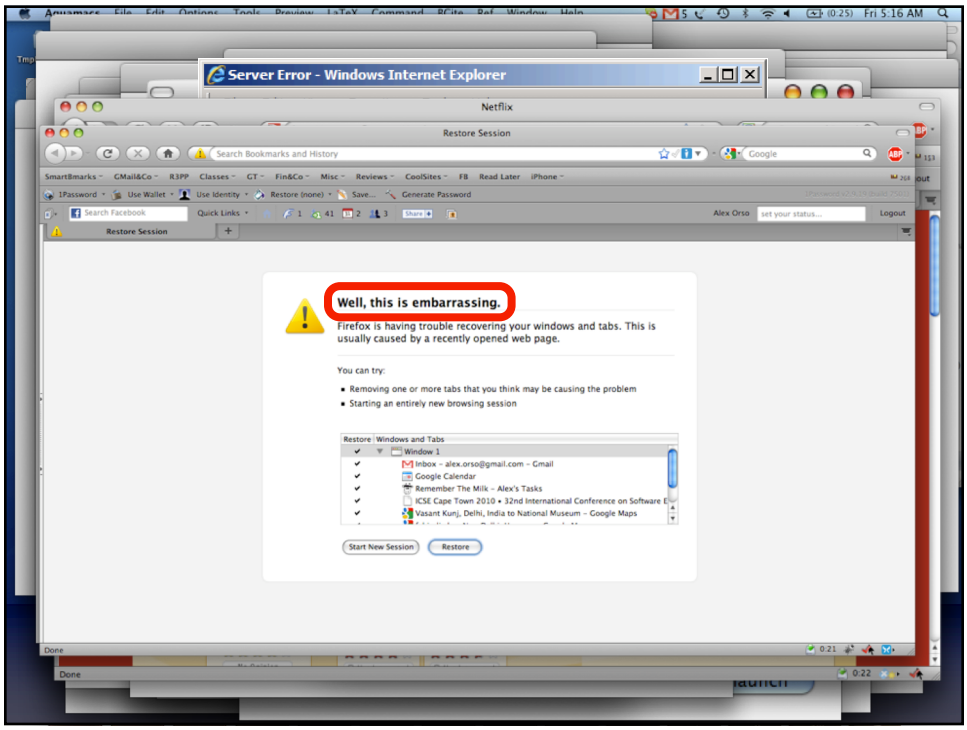
Producer view

- Fulfillment of contractual obligation
  - System conforms to the user/customer specifications
- Adherence to software process and satisfaction of customer's expectation
- Usability, modifiability, maintainability, portability, etc.

Correctness

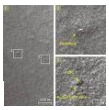
- Commonly, quality is associated with correctness
- The software runs with few if any "problems"
- during the operation
  - if a problem does occur, it has minimal impact to the operation


The application SystemUIServer quit unexpectedly.
2008-07-02 03:08:32 -0400
EXC_BAD_ACCESS (SIGBUS)
KERN_PROTECTION_FAILURE at
0x0000000000000000
Thread 0 Crashed:
0 objc_msgSend + 28
1 prefsNotify + 248
2 rlsPerform + 103
3 CFRunLoopRunSpecific + 3141
4 CFRunLoopRunInMode + 88
5 RunCurrentEventLoopInMode + 283
6 ReceiveNextEventCommon + 374
7 BlockUntilNextEventMatchingListInMode + 106
8 _DPSNextEvent + 657
9 -[NSApplication nextEventMatchingMask:untilDate:inMode:dequeue:] + 128
Ignore
Report...
Relaunch


Server Error - Windows Internet Explorer
Netflix
Restore Session
Well, this is embarrassing.
Firefox is having trouble recovering your windows and tabs. This is usually caused by a recently opened web page.
You can try:
Removing one or more tabs that you think may be causing the problem
Starting an entirely new browsing session
Start New Session
Restore

Spectacular Software Failures

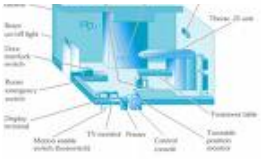
- NASA's Mars lander: September 1999, crashed due to a units integration fault



Mars Polar Lander crash site?

- THERAC-25 radiation machine : Poor testing of safety-critical software can cost *lives* : 3 patients were killed
- Ariane 5 explosion : Millions of $$
- Intel's Pentium FDIV fault : Public relations nightmare

THERAC-25 design



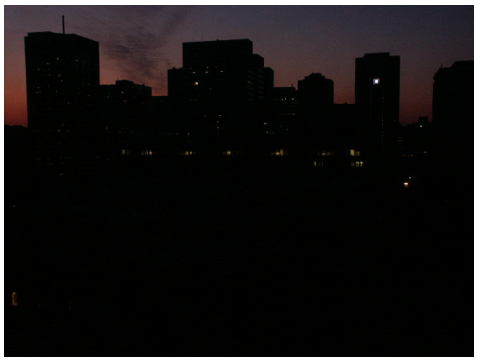
Northeast Blackout of 2003

508 generating units and 256 power plants shut down

Affected 10 million people in Ontario, Canada

Affected 40 million people in 8 US states

Financial losses of $6 Billion USD



The alarm system in the energy management system failed due to a software error and operators were not informed of the power overload in the system

Software is everywhere!!

Why is so difficult to build software?

- Complexity
  - No linear relation with size
  - Technical and management problems
- Conformity
- Changeability
- Invisibility

Solution

- High level languages
- Time-sharing
- Unified programming environment
- …

There is no silver bullet

Costly Software Failures

- Huge losses due to web application failures
  - Financial services : $6.5 million per hour (just in USA!)
  - Credit card sales applications : $2.4 million per hour (in USA)
- In Dec 2006, *amazon.com's* BOGO offer turned into a double discount
- 2007 : Symantec says that most security vulnerabilities are due to faulty software

Software is Buggy!

- On average, 1-5 errors per 1KLOC
- Windows 2000
  - 35M LOC
  - 63,000 known bugs at the time of release
  - 2 per 1,000 lines

For mass market software 100% correct is infeasible, but we must **verify** the SW as much as possible

Failure, Fault, Mistake

Failure

Observable incorrect behavior of a program. Conceptually related to the behavior of the program, rather than its code.

Fault (bug)

Related to the code. Necessary (not sufficient!) condition for the occurrence of a failure.

Mistake

Cause of a fault. Usually a human error (conceptual, typo, etc.)

Failure, Fault, Mistake: Example

```
1. int double(int param) {
2.     int result;
3.     result = param * param;
4.     return(result);
5. }
```

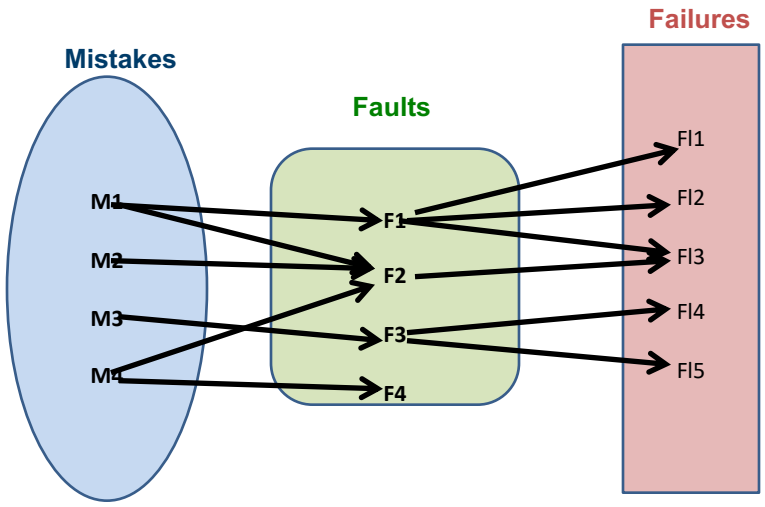
A call to double(3) returns 9

Result 9 represents a failure

Such failure is due to the fault at line 3

The error is a typo (hopefully)

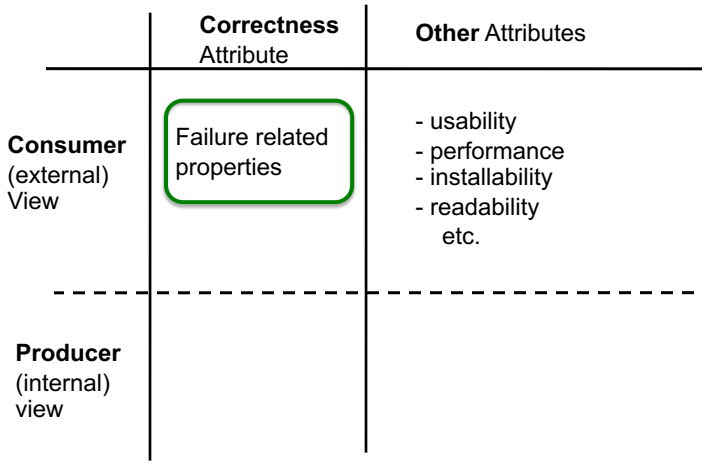
Relation among mistake/fault/failure



Back to Consumer/Producer Views

	Correctness Attribute	**Other** Attributes
Consumer (external) View		
Producer (internal) view		

Back to Consumer/Producer Views



"Correctness" centric properties

- Direct failure
  - Number of failures
  - Distribution of failures (by functional area; by types, etc.)
  - Defect source classifications
- Failure likelihood and reliability
  - Failure probability
  - Mean-time between failures
- Failure severity and safety
  - Failure fix (by recovery cost/length)
  - Failure impact (by user work arounds)

Back to Consumer/Producer Views

	Correctness Attribute	**Other** Attributes
Consumer (external) View	Failure related properties	- usability - performance - installability - readability etc.
Producer (internal) view		

Back to Consumer/Producer Views

	Correctness Attribute	**Other** Attributes
Consumer (external) View	Failure related properties	- usability - performance - installability - readability etc.
Producer (internal) view	Fault related properties	- design complexity - size - maintainability etc.

Finer grain view

- Software Fault : A static defect in the software
- Software Failure : External, incorrect behavior with respect to the requirements or other description of the expected behavior
- Software Error : An incorrect internal state that is the manifestation of some fault

Faults in software are equivalent to design mistakes in hardware.

Software does not degrade.

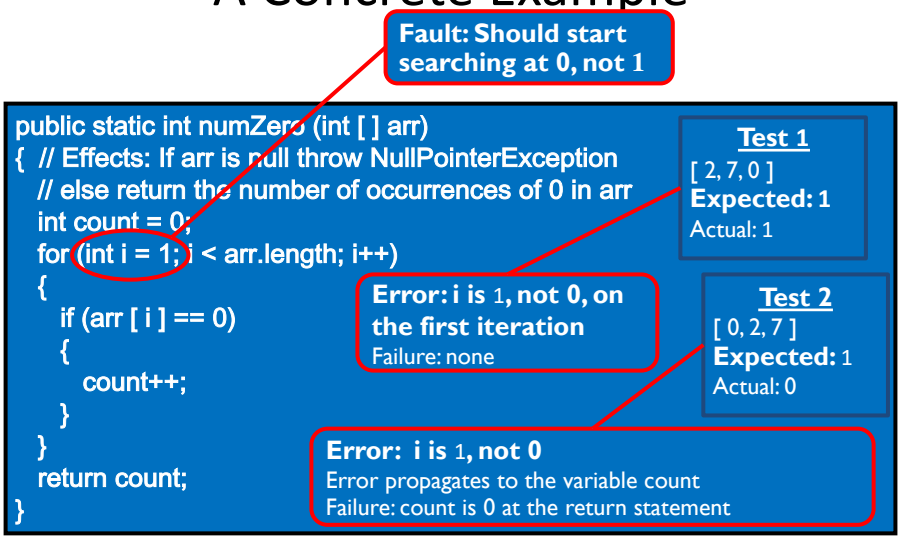
Example

- A patient gives a doctor a list of symptoms: failures
- The doctor tries to diagnose the root cause, the ailment: fault
- The doctor may look for anomalous internal conditions: errors

Most medical problems result from external attacks (bacteria, viruses) or physical degradation as we age.

Software faults were there at the beginning and do not "appear" when a part wears out.

A Concrete Example



Quality = dealing with defects

- Prevention
  - Requirements
  - Model analysis
- Detection and removal
  - Inspection
  - Testing
- Containment
  - Monitoring

A general view (from our perspective)

- The software system must do what is required/expected.
  - Must do the "right" things
  - (Do not perform the wrong functions!)
- The software system must do the required functions correctly.
  - Must do the things "right"
  - (Do not perform the intended functions incorrectly!)
- …

Approaches to Verification

- Testing: exercising software to try and generate failures
- Static verification: identify (specific) problems statically, that is, considering all possible executions
- Inspection/review/walkthrough: systematic group review of program text to detect faults
- Formal proof: proving that the program text implements the program specification

Comparison

Testing
- Pros: reveal failure
- Limits: small subset of the domain (=> risk of inadequate test set)

Static verification
- P: consider all program behaviors (and more)
- L: false positives, may not terminate

Review
- P: systematic in detecting defects
- L: informal

Proof
- P: prove correctness
- L: complexity/cost (requires a spec)